

UNCLASSIFIED

Defense Technical Information Center
Compilation Part Notice

ADP010977

TITLE: Planning for Change With A Holistic View of the System

DISTRIBUTION: Approved for public release, distribution unlimited

This paper is part of the following report:

TITLE: Strategies to Mitigate Obsolescence in Defense Systems Using
Commercial Components [Strategies visant a atténuer l'obsolescence des
systemes par l'emploi de composants du commerce]

To order the complete compilation report, use: ADA394911

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, etc. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:
ADP010960 thru ADP010986

UNCLASSIFIED

Planning For Change With A Holistic View Of The System

Ted Dowling

Systems & Software Engineering Centre
Defence Evaluation & Research Agency (DERA)
St Andrew's Road
Malvern
WR14 3PS
UK
+44 (0) 1684 894734
ejdowling@sec.dera.gov.uk

Summary. The example of obsolescence which perhaps comes most readily to mind is that of electronic components that are no longer available. However, this is just a special case of the more general form of obsolescence that arises when a system no longer provides an adequate solution to a user's problem. This may arise because the problem has changed or because the solution (the system) has, in some way. In practice, both the problem and solution are changing continuously and asynchronously. The approach to obsolescence management proposed here depends on recognising and planning for this change. In essence, it involves looking forward to how the demands on the system and the technology that provides its capability may both change. Simulation is a crucial tool in doing this. In the light of the understanding of expected changes, the design of the current system is arranged to facilitate transition to the modified system and a change plan is produced. This paper also looks briefly at the impact of the proposed approach on the broader system engineering activities and the commitment it requires from the system's customer.

Background to Paper. The Defence Evaluation and Research Agency (DERA) is the prime source of research for the UK Ministry of Defence (MOD), and also provides a major source of independent advice to MOD during all stages of systems procurement.

The Systems and Software Engineering Centre (SEC) is a relatively new body within DERA, being established in 1994 to act as a focus for professional software (and soon after, systems) engineering within DERA. The majority of its complement of about 260 staff have an industrial background.

The SEC has responsibility for the systems and software standards and practices used across DERA (which has a staff of around 11,000). It provides the editor for the draft ISO standard (ISO15288) on systems engineering and is influential in setting the systems engineering direction of MOD's procurement arm, the Defence Procurement

Agency (DPA). It provides systems and software engineering support to a wide range of programmes within DPA. The SEC is also leading in the field of capability assessment and evaluation, eg in developing and applying various Capability Maturity Models (CMMs). The author is the SEC's Technical Manager.

Despite this background, it should be made clear that this paper does not constitute the results from a MOD-funded research programme, nor does it represent the official view of MOD, DERA or the SEC. Rather it captures the personal views and thinking of the author. However, the author is pleased to acknowledge the rich source of ideas he has encountered in the SEC, DERA, MOD, Defence Scientific Advisory Council (DSAC) working parties and other contexts.

Introduction. Obsolescence happens because the world changes. Today, this change happens more and more rapidly. Sometimes the change is predictable (such as the increase in power of processor chips), sometimes it is rather more unexpected and of a more dubious nature (eg, to take a completely different domain, the disruption caused by the rapid rise - and sometimes rapid fall - of "dot com" companies on the stock market).

Defence systems exist in this volatile world and yet in many ways are antithetic to it. They have a long "gestation" period and are expected to be in use for extended periods. It is clear that a way to mitigate the impact of changes is required.

Many approaches are possible, all of which make some contribution. Well known techniques include attempting to create system architectures in which components can easily be replaced when appropriate, through concepts such as modularisation, layering, fixed and open interfaces, and standardisation.

This paper considers a complementary approach based on simulated "virtual" systems. It is a generic approach that supports, but is not restricted to, the particular problem of managing obsolescence in electronic components.

The Nature of Obsolescence. It is helpful to consider some basic questions:

- What is obsolescence?
- What becomes obsolete?
- Why do things become obsolete?

Obsolescence. Obsolescence is the act of becoming obsolete. The dictionary defines obsolete as "no longer functional". However, we can extend and clarify this by considering that an item is obsolete when both of the following are true:

- It no longer meets the user's need (we assume it once did!)
- It is not possible to make it do so without considerable effort – if at all

A very simple case of failing to meet the user's need occurs when an item ceases to function, and a simple reason for not being able to remedy this in an easy way is if the item is no longer available. This is the classic electronic component obsolescence situation, and is perhaps the easiest to consider, but it is far from being the only way in which obsolescence can occur.

In many cases, obsolescence is a gradual process. As time passes, it may well be that the item diverges more and more from what the user needs and at the same time it becomes more and more difficult to bridge this gap.

It is also worth noting that an item can be obsolete in one context (eg in respect to one user's needs) while not being so in another.

What become obsolete? It is important to note that obsolescence strikes at all levels, from the smallest (electronic) component to a complete system. Clearly, if a component becomes obsolete, so often does the (sub)system of which it forms a part, but equally a system can become obsolete while each of its constituents remains current (in some context at least). If the collection of components and their interaction no longer provide the functionality and performance required, and it is not simple to change or directly replace them, then the system is obsolete.

Hardware components can become obsolete because they are no longer available and cease to provide the necessary features, either through failure or because more is now needed of them than originally. COTS software items too can become obsolete in the same sort of way (although failure is less likely). However, bespoke software can also be obsolete if changing it, while possible in theory, becomes too difficult, costly and risky to be worthwhile.

Why do items become obsolete? Perhaps the most obvious cases of obsolescence occur within electronics. Anybody who owns a PC at home is

only too aware that even a top-of-the-range machine purchased three years ago is now likely to be considered out of date, with little residual re-sale value. It may still be possible to do most of what is required of it, but now very slowly by today's standards. Virtually every item (processor, bus, memory, disk, CD drive, etc) has seen significant enhancement over the period. In some cases, there are new capabilities that are just not available on the "old" machine (eg DVD).

In a lot of ways, though, the machine is not obsolete because it lacks a fundamental capability, but because it lacks enough of what it does have (not enough processor power, not enough RAM, not enough disk space, not enough graphics speed, etc). Furthermore, while in principle most of these aspects could be upgraded, the cost would comfortably exceed the price of a brand new replacement.

And why is what was enough three years ago no longer sufficient? Largely because expectations have increased - the expectations of the end user and the expectations of the software writer, who now assumes a basic configuration that is valid today but was not so three years ago. It is interesting to note that this software is a COTS item - so COTS is helping create obsolescence not prevent it!

Systems can also become obsolete because they simply do not provide the functionality that is required in a changing environment (if they ever did!). Most changes in environment that cause obsolescence are gradual; the change is continuous. However, some changes are much more abrupt. Betamax home video recorders became obsolete very rapidly once the VHS-Beta format battle was lost, for example.

In addition, systems become obsolete simply because failures (primarily in hardware, but software can be affected too) happen and there is no reasonable source of spares with which to effect a repair.

The poor owner of the PC and the video recorder is totally powerless to prevent his systems being made obsolete by external, "wide world" forces over which he has no control. The best he can do is aim to predict correctly where the future is leading (eg VHS) and take reasonable steps to ensure he can follow (eg ensuring upgrade potential in his PC, such as spare card slots and bays).

Obsolescence in the defence world. Of course, these same pressures and issues apply to defence systems. They too become obsolete for two basic reasons:

1. The environment in which the system acts has changed in such a way that it can no longer offer adequate performance

2. The system is subject to faults that can no longer be repaired easily because of a lack of suitable spares/skills/facilities

Again, since the defence world is ever-less-important on a global scale - particularly in the most rapidly changing areas such as computing and communications - the obsolescence may be increased by COTS items.

Naturally, the procurers and owners of systems - like the PC/video buyer - attempt to minimise these risks. However, the emphasis is often on the initial procured system and some rather general upgrade capability (eg not consuming more than 50% of the processor power), rather than on more detailed forward planning.

It is not suggested here that the future is currently ignored when procuring a typical system, or that consideration of the future does not get reflected in non-functional requirements such as for extensibility. However, the approach outlined here does perhaps differ from that widely adopted in its emphasis on:

- A broad view of the future that encompasses the physical system, the user, the method of use, etc
- An in-depth (at least to the degree that is appropriate) exploration of the future
- Explicit capture and maintenance of the future-oriented material

Planning for Change. Obsolescence is caused by change, and its impact can only be reduced by anticipating and accommodating change. Change is natural and inevitable, and it is futile to ignore it. Procurement approaches that are predicated on fixed and detailed up-front system specifications, rigid fixed-price contracts, and a fear of so-called "requirements creep", come close to emulating King Canute¹.

Rather, the need is to recognise change and cater for it from the start. This change will arise from a number of distinct sources:

- The world in which the system is to operate is ever-changing. What the user needs to be able to do, and consequently, what he wants the system to do for him, will change - perhaps slowly, perhaps rapidly.
- The technologies available for the system to exploit will change (for the better) and what

was previously impossible/impractical will become feasible.

- The user's perception of what he wants of the system will change from the very moment it is in use, even if the rest of the world were static. Only when the system is used for real will users identify additional or different features they desire.

The first two points can be addressed by actively exploring how the possible problems (the first point) and the possible solutions (the second point) might change in the future. This is discussed further below.

The last of these points is almost a separate issue. It is what makes systems developments based on paper specifications and paper interim products (design specifications, etc) inherently weak. It is best addressed by a development in which end-user involvement is as deep as possible throughout; there is great emphasis on increments and iteration; and there is maximum flexibility to change direction. In the software world, disciplined RAD (Rapid Application Development) methods such as DSDM (Dynamic System Development Method)² provide such a development technique.

Predicting Change. We have a number of sources that can help us identify changes in both the problem and solution domains, eg:

- The commercial world (which is often only too ready to promote "futureware"!))
- Research programmes, both general and defence-oriented
- Military intelligence

COTS items are likely to be especially suitable for this "crystal ball gazing" since their developers and suppliers usually have a well-defined forward plan for future products.

Some changes are in fact very predictable, especially in the solution domain. We know that processors will become more powerful, communication bandwidth will increase, mobile 'phone technology will become ever-more sophisticated (eg internet access), and so on.

Of course, the solution and problem domains are by no means disjoint. One impact of COTS is that potential foes are likely to enjoy essentially the same access to COTS items as we are. Indeed, it may be that they are much more agile in exploiting them than some national defence forces. Hence a potential solution may also be a potential problem.

¹ A Viking king who commanded the waves to stop coming up the beach (although in fact he did not actually believe he could control the waves, but wanted to show that mortals are powerless over some things).

² See www.dsdm.org

It is also important to consider less obviously predictable changes. By definition, these are more difficult to identify, but "what if" scenarios based on the more outlandish of the concepts pursued in research environments should not be ignored.

The usual combination of "likelihood of happening" and "impact" can help guide the choice of possible changes for further consideration.

Managing Change. Combating obsolescence requires relevant possible changes to be studied, so as to influence the system as a whole (its design, concept of use, etc) throughout its life.

For example, we can consider a command and control system in which data exchange bandwidths are much greater than is currently achievable, but which might reasonably be expected to be attainable just a few years after initial delivery of the system.

It might be that totally new opportunities for the way in which the system is used are opened up by this increase in capability. Perhaps the user could have more or better (eg more accurate) information available in the same time, perhaps he could just have the same data but much more quickly, or perhaps more people could have the same data. Any of these alternatives might suggest a different way in which the system might be used.

Other examples might be: i) future technology makes equipment so much more portable that each soldier can carry what now goes in a vehicle; ii) many more users need to be connected simultaneously; iii) the enemy develops a more powerful jamming capability. All these could make the current system obsolete, even if obsolescence in the sense of component availability is not an issue at all.

At any point in time, therefore, we have the following entities to consider:

1. The problem space³ - the environment in which the system is to be used and from which user needs emerge. This is many-faceted, covering the full spectrum from physical terrain and physical platforms to knowledge and tactics of all participants other than the system operator.
2. The solution space - the physical system itself and the way in which it is used:

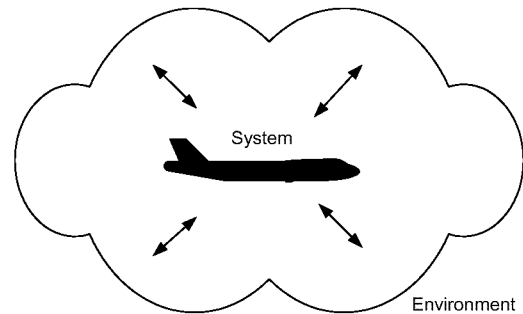


Figure 1 The System in its Environment

Furthermore, by looking ahead, we have two or more such pairs:

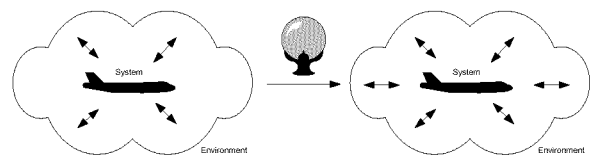


Figure 2 The System Now and in the Future

The future view represents the anticipated system and its use. This vision of how the system will be required to evolve forms a key input into how it is designed now. Knowing that a system and/or the way it is used will change in a particular way in the future is a crucial piece of data to inform the system design.

Very broadly, we have a number of inter-related aspects to consider:

1. The user needs within an environment now
2. The future user needs within a future environment
3. The system and its use that meets the needs now
4. The future system and its future use

There are a number of levels of abstraction at which we can consider all these items: the problem and solution domains, the user needs and the system that meets them, and the system's requirements and design. We can also consider "the system" to be the physical system, the users, the method of use, etc. These various aspects are related as shown below:

³ Note that here the "problem space" is not the collection of problems, but the context in which the problem exists and in which the system aims to provide a solution.

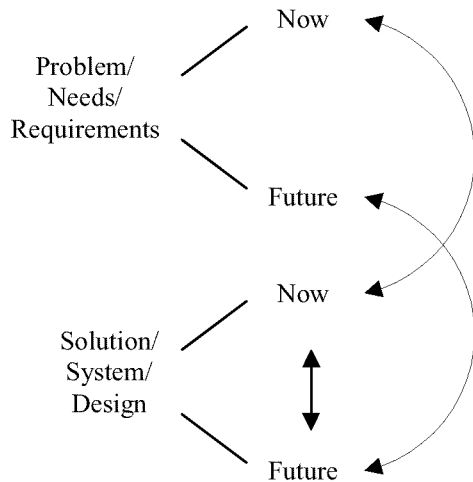


Figure 3 Solution and Problem Interactions

The design of the system that is produced now is, of course, driven by the requirements, but in addition - especially in a COTS-based system - the requirements are tempered by what is possible in the design and trade-off is needed. Similarly, when considering future needs and system possibilities, the same relationship between requirements and design exists.

Also, the system that we design for the future has an influence on how we design for the present, so that the transition to the new system is facilitated. On the other hand, we need to consider the current design when deriving the future system, for the same reasons.

Of course, it may be that in considering the future we decide that the gap between the current system and the one that is appropriate for the future is so great that a continuous transition is not appropriate and a better option is to develop a system with a short life and completely replace it in the future.

A number of forward-looking horizons may be appropriate. For example, we might look at now, 5 years' time and 10 years' time and consider how the problem and solution might appear at each stage, and how to accommodate this. Obviously, the further into the future the view is taken, the more approximate are likely to be the various items of information.

In terms of the system engineering artefacts that must be created, managed, etc, this approach introduces a number of new items, in addition to all the classic ones that exist when no forward look is taken:

- The requirements for the future system
- The design for the future system
- A change plan for the transition from the current system to the future

The change plan sets the way forward for the system based on the predicted changes in technology etc (the solution space) and needs (the problem space). It may include interim stages along the path from the current to the future positions, depending on how large the current-future gap is.

As with classic "point"⁴ system design, traceability between the design drivers and design features is important. Thus, for example, it is crucial to maintain traceability from a particular design aspect back to its justifying element of the change plan.

The forward-looking artefacts discussed here clearly need to be maintained as time passes. Periodically, the assessment of future needs, future solution options (eg new technological capabilities) and the design for the future system itself can be revisited and updated as appropriate, resulting in a revised change plan.

Thus while the system itself may be essentially static (ignoring routine fixes and minor enhancements), the future system – that is, the envisaged actual system, the way it is used, etc – may be "upgraded" more frequently.

The following diagram shows successive versions of the physical and future system with asynchronous upgrades. The future system bars show the lifetime of various versions of the *prediction*, not of the actual system. Thus, for example, version 3 of the future system which is current when the physical system is upgraded to version 2 may predict the position some years after version 2 comes into service. Version 3 of the future system – ie predicted future needs and system design – will influence version 2 of the physical system via the relevant change plan, but it is not necessarily true that the introduction of a modified system will change the prediction for the future, so the future system is not affected. What must be upgraded, of course, is the change plan.

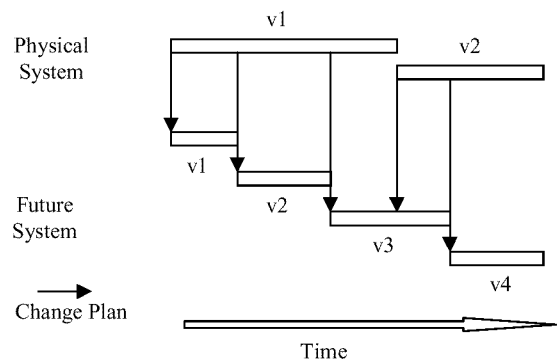


Figure 4 Physical and Future Systems

⁴ ie that addresses the problem and solution at just one point, not across a now-future range

Note that the change plan is updated whenever either the physical or future system is modified.

Since in practice the physical system is unlikely to be totally static, the work on revising the future system can inform and influence minor changes to it.

The requirements and design for the future system and the change plan are products of an obsolescence management activity, controlled by an obsolescence management plan. Related activities to be covered by the plan include identifying the parts of the system – or problem domain – that are likely to be affected by obsolescence and deciding at what frequency to produce new versions of the future system.

Simulation. Simulation of various kinds (including here, for convenience, modelling) is a well-established tool to assist in the development of defence systems. Analysis activities, such as support for balance of investment decisions, rely heavily on simulation to explore the cost-effectiveness of various system options. More generally, simulation-based acquisition is achieving growing acceptance and importance, allowing a whole range of alternatives to be explored during system design and to be validated during system integration and acceptance. However, simulation specifically to address obsolescence issues appears to be relatively rare.

Simulations that represent the system as it is currently designed, and of the environment with which it interacts, are required to assist the understanding of interfaces, performance, emergent properties, etc during design, and to aid integration and validation.

In addition, simulation is an obvious way (indeed, probably the only way) to explore the system and its environment in the future.

For designing today's system, fine-grain, high fidelity simulations may be needed, but the more one is looking into the future, the more likely it is that coarse-grain, low fidelity simulations will be appropriate. Since such simulations are generally quicker and cheaper to develop, this has the advantage of making it more feasible to explore a number of different variants of the predictions.

“Broad brush” simulations at a relatively high level of abstraction may well be used during the initial stages of system development anyway (eg in exploring user needs and in identifying options).

With suitable forethought the same simulations may be exploitable for looking at future systems, for example through parameterisation.

Systems Engineering Impact. The approach described here introduces a number of new systems engineering artefacts:

- Obsolescence management plan
- Change plan
- Future system requirements
- Future system design
- Future simulations (system and environment)

All these require to be seen as part of the core set of systems engineering artefacts for the system and to be managed appropriately.

In addition, we can see how this approach affects the systems engineering activities. One obvious impact is that when the current system changes in some way, all these new artefacts must be examined and refined as appropriate, with configuration management applied. Traceability is also a key concern.

The new (draft) ISO systems engineering standard, ISO15288, identifies a number of processes, as shown in Figure 5.

It is clear that obsolescence management has an impact on most of these to a greater or lesser extent and in one way or another. Considering future requirements and designs as well as current ones inevitably introduces additional work and complexity, which affects processes across the board. However, the major impact is on Stakeholder Needs Definition, Requirements Analysis, Architectural Design and Implementation.

Stakeholder Needs Definition is concerned with understanding what the system must do, and obsolescence management extends this to considering future needs as well as current/short-term ones. The future requirements will be identified here and this activity will require appropriate simulations of the future problem space.

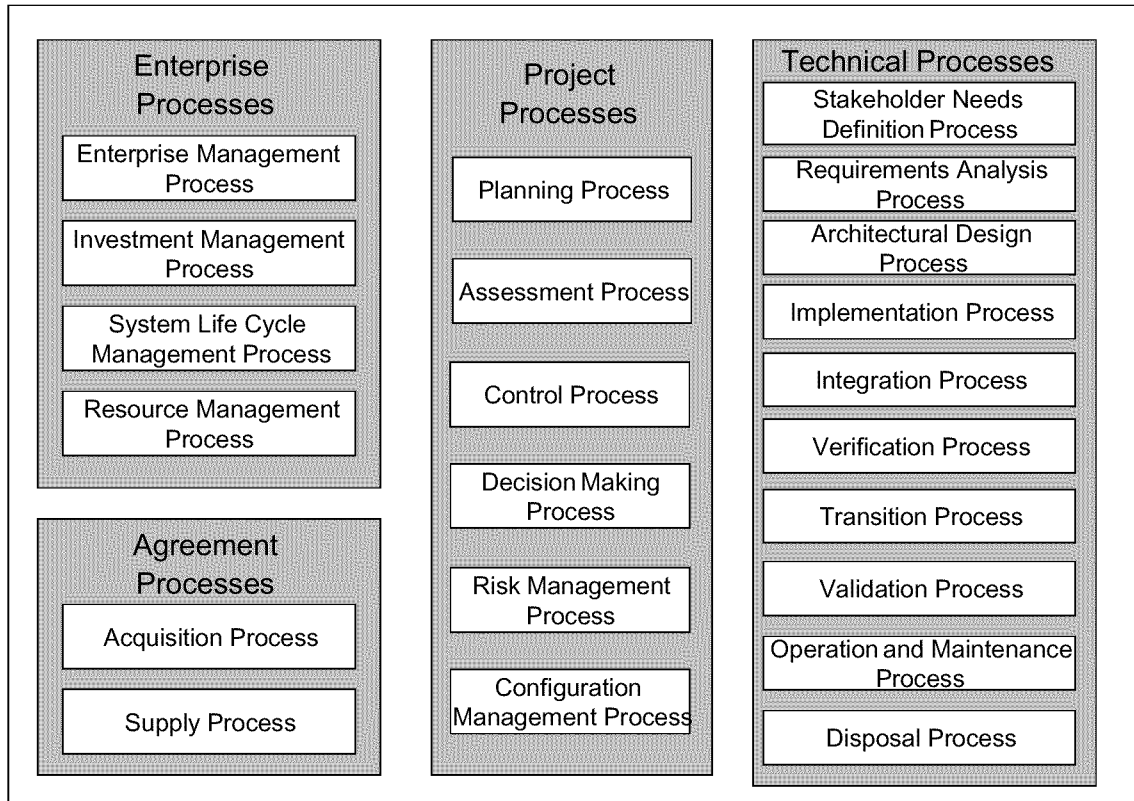


Figure 5 – ISO15288 Systems Engineering Processes

Requirements Analysis leads to a system requirement based on the stakeholder needs. In practice, there is often a rather hazy line between requirements analysis and design since often a particular model (or several models) of how the system might look tends to emerge at this stage. Hence the impact of future solutions may well need to be considered here, as well, of course, as considering future needs in addition to current ones.

Architectural Design is obviously very much affected by the need to consider what solutions might exist in the future to cater for the identified future requirements. Architectural Design involves trade-off decisions between various design options, and this is a key activity when deciding how the current design should be influenced by obsolescence management issues. It is here that the future design is derived, using appropriate simulations. The change plan will also be produced here.

Implementation is concerned with taking the output of the Architectural Design process as a set of requirements for lower level sub-systems and repeating the analysis and design activities. In practice, for large systems such as an aircraft, it may well be that it is at this stage that many obsolescence issues are first studied in depth. However, it is important that their impact is

reflected upwards. For example, it may be that during the Implementation activity, it is decided that a particular box will be half its current size and weight in five years' time. The future aircraft design must reflect this opportunity.

The ISO standard is clear that the various processes are not necessarily executed sequentially. Even ignoring obsolescence, iteration between the four processes discussed here is vital, especially when COTS is being exploited. The approach described here can be seen as introducing a parallel iteration between requirements and design for the future system, and between the current and future systems, as shown in Figure 6.

It is interesting in passing to note that while the ISO standard certainly does not preclude obsolescence management as described here, it makes no explicit mention of catering for it. Its focus is on maintaining the system as first delivered and reacting to new needs as they arise, rather than predicting new needs and solutions. It is reactive rather than proactive.

Procurement Impact. A very obvious impact of this approach is that it involves extra effort, cost and time, compared with simply ignoring obsolescence. This is a major issue since it seems

all too common that, for a variety of reasons, investment in “up front” activities for systems is difficult to obtain.

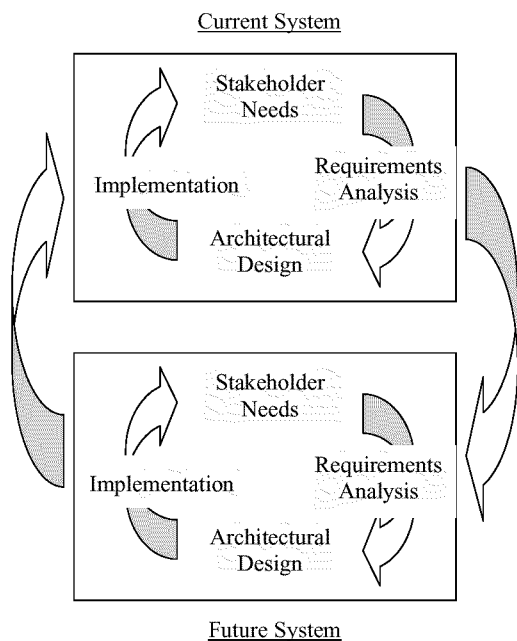


Figure 6 Iteration Within and Between Systems

The amount of effort that it is appropriate to put into obsolescence management clearly depends on the risk of obsolescence and its expected impact. In this way, obsolescence is no different from any other factor influencing the system. The overall risk management for the system should include assessing obsolescence risks and deciding upon the appropriate degree of forward planning. However, it is clear that the necessary effort could well be significant.

Since obsolescence arises from the problem domain as well as the solution domain, this is obviously an issue that should be considered by the user/procuree at a very early stage. It is not driven solely by aspects of equipment obsolescence and cannot be considered as something to be left to the system supplier alone. The approach adopted may have a major impact on the system's through-life cost profile.

Neither is it a matter simply of cost and possibly timescales. It may be that analysis shows that to address an anticipated obsolescence problem, the initial system should have characteristics that would be considered sub-optimal if the system were not to be upgraded. Thus initial users might be asked to accept sub-optimal performance now to provide a better (or perhaps simply cheaper) system later – based on predictions of future needs and

solutions. There are obviously very complex trade-offs and decisions to be made!

Conclusion. Obsolescence in systems has many causes, but ultimately is due to change in the problem space and/or the solution space. By attempting to understand the nature of this change for any given system, we can facilitate adapting to it. This requires the future system requirements and design to be derived, and a change plan to transition from the current to future system to be produced. COTS elements may be particularly amenable to this kind of forward looking since they often have a predictable development path.

Obsolescence must be a major element of the system's risk management and this will decide the degree of investment that is appropriate. There may also be major issues involved in trading off immediate functionality to facilitate future changes. Procuree commitment to this approach is therefore vital.

Simulation will play a major role, especially in assessing future needs and solutions. These simulations and the various other artefacts (future design, etc) become key systems engineering products and must be managed accordingly. The "whole" system becomes the traditional physical system, its design, etc plus these other items.

It is clear that this approach is non-trivial. However, to at least ask for all systems the question “how much of this should we do?” seems to be vital in reducing the impact of obsolescence.